Application Note

CORE MODULE @ 13,56 MHz

ACMx/APPx – Series

Software Description
Firmware: ACMx / APPx

ARYGON TECHNOLOGIES

Erthalstrasse 1
D - 55118 Mainz
Germany
Phone +49 (0) 61 31-30476-0
Fax +49 (0) 61 31-30 476-20
info@arygon.com • http://www.arygon.com

Product Name: ACMx / APPx

Document: Application Note Software Description

Author: A. Kretschmann

General Version: 2.0

Document History

Revision	Date	Description
0.1	April 2005	First published version
0.2	July 2005	Added: (µC Firmwareversion V0.5): Bootloader function, µC sleep Mode, copy value block command, halt command, Mifare® scenario.
0.3	October 2005	Added: (µC Firmwareversion V0.6): RFConfiguration command, µC EEPROM commands, µC I/O Port commands, party line (RS4xx) commands.
0.4	November 2005	Added: analog input port command.
0.5	December 2005	Some small errors corrected.
0.6	January 2006	Reader serial number added.
1.0	2006.01.18	Release Version.
2.0	2006.05.05	Stand-alone functionality with Wiegand™ interface added.

Contact Information

For additional information and sales office addresses visit:
http://www.arygon.com or **http://www.nfc-global.com**

Contents

1. Conventions and notations

1.1 Representation of numbers

0xA3	Hex notation. e.g.: 0xA3 = A3 hexadecimal.
'123ABCD'	ASCII notation. '123ABCD' (= 0x31 0x32 0x33 0x41 0x42 0x43 0x44)
CR	Carriage Return = 0x0D
LF	Line Feed = 0x0A

1.2 Abbreviations

ACMx	ARYGON Core Module (Variant x)
APPx	ARYGON Plug and Play Module (Variant x, incl. ACM)
HSU	High Speed UART (Baud rates above 115 kBaud).
GPIO	General Purpose digital Input or Output.
NFC	Near Field Communication. (ECMA 340)
UART	Universal Asynchronous Receiver Transmitter = RS232 in CMOS/ TTL.
µC	Optional equipped Microcontroller on the ACM Board.
TAMA	Name of the Philips digital / analog Front End IC (PN531).
RFU	Reserved for Future Use.
PWM	Pulse Width Modulation.
ADC	Analog Digital Converter.

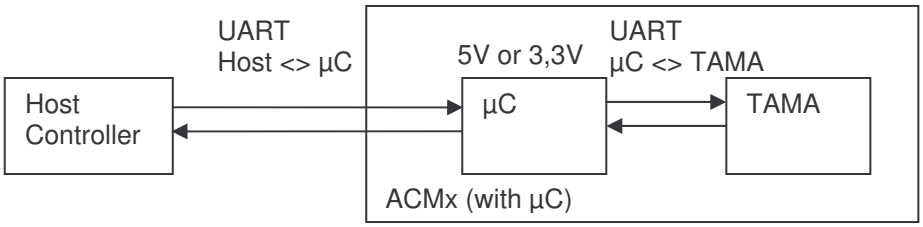
2. Host interfaces

2.1 UART (HSU) interface

After power on data are transmitted at 9600,n,8,1, no hardware handshaking as default.
The baud rate is adjustable both between Host Controller <> Reader µC and between Reader µC <> TAMA.
In principle it is possible to select a different baud rate between µC <> host and µC <> TAMA. To avoid buffer overrun errors inside the µC for higher amount of data (peer to peer or T=CL datapackets) it is mandatory to select the same baud rate on both sides.
Different baud rates can be selected for small datapackets (e.g. ISO14443A-3 or special µC commands).
Possible adjustable baud rates on both sides are:

9.6 **19.2** **38.4** **57.6** **115.2** **230.4** **460.8** kBaud.

(The fastest baud rate of 460.8 kBaud is only possible with the 5V µC which runs with a higher clock).



2.2 APPx high speed RS232 interface

The normal UART Software-interface protocol is used, no difference from software side.

2.3 APPx high speed RS4XX interface

The APPx board provides optional RS485 in half or full duplex or RS422 in half or full duplex.
In partyline mode the host has to poll for the µC responses to avoid bus collisions.
Details can be found in the polling command description ('aplxx').

2.4 SPI interface

- TAMA without µC, refer to PN531 User Manual for details.
- µC with SPI interface, is planned.

2.5 ACMx with I2C interface

- TAMA without µC, refer to PN531 User Manual for details.
- µC with I2C interface, is planned.

2.6 ACMx with USB interface

The normal UART software-interface protocol is used, no difference from software side. The USB interface driver emulates a Com Port on host side (VCP Virtual Com Port interface).

2.7 APPx with USB interface

The normal UART software-interface protocol is used, no difference from software side. The USB interface driver emulates a Com Port on host side (VCP Virtual Com Port interface).

2.8 Wiegand™ Interface

Stand alone µC firmware necessary.

3. Philips protocol (no µC equipped)

Refer to Philips PN531 User Manual (firmware description), *UM0501-xx.pdf.*

4. ARYGON protocol (µC equipped)

The availability of the µC is *mandatory* in case of:

- Stand-alone applications like access control.
- Wiegand™ interface.
- Party line applications (RS485).
- High level language usage in any terminal program (for easier testing).
- High level language usage in Mifare® applications where checksum handling is not required or not possible because of host limitations.
- Personalization of the Reader. Only with µC equipped it is possible to store the Mifare® authentication keys into the Reader (µC EEPROM). TAMA provides no user EEPROM.
- More than two user GPIO pins needed for special digital switching or digital read functions. (TAMA offers two user GPIOs, µC offers 9 user GPIOs, a PWM output pins and an analog input sold pad.)
- Performing a TAMA hardware reset via a software command from host to µC.

4.1 Protocol modes

Several protocol modes are supported via a **mode select command byte** which is sent as first Byte in every data packet from host to the µC. The mode select command byte tells the equipped µC how the following bytes have to be interpreted.
The mode select command byte is always sent in **ASCII format**.
Following mode select command Bytes are available:

'**0**' High level language in ASCII format.
(Common µC commands and Mifare® commands)

'**1**' High level language in Binary format with addressingbyte for party line.
(Common µC commands and Mifare® commands)

'**2**' Philips protocol (TAMA language) in binary format.

'**3**' Philips protocol (TAMA language) in binary with addressingbyte for party line.

Generally, after sending of a command to the µC, the host has to **wait** for the complete µC response before the next command can be transmitted!
This is essential for mode switching within a command sequence because it is allowed to send new commands in another mode than the previous command was sent.

4.2 ASCII format (mode select command byte = '0')

This protocol was designed for easy handling. Data are transmitted as ASCII hexadecimal which can be sent and displayed on any terminal program for test purposes.
In this mode there is no checksum performed. The response is sent in ASCII format from the µC to the host as well.

The ASCII protocol is available for common µC and Mifare® commands.

ASCII format example:
Host to reader'0ah02' *// set baud rate to host side to speedlevel 2 (38,4kBaud).*
Reader to host 'FF000000'CRLF *// µC response packet. Status = OK.*

4.4 Binary format (mode select command byte = '1' or '2' or '3')

This protocol should be used for serious industrial data communication to use the advantage of the checksum handling.

With mode select = '**1**' the µC performs a checksum handling for the ASCII high level language and optional party line (RS4xx) is provided.
If the reader ID is set to 0x00, then the host can use the checksum handling **without** the party line functionality. No µC response packet buffering and host polling are necessary.
If the reader ID is set greater than 0x00 then party line is **activated** and host polling is mandatory.

With mode select = '**2**' or '**3**', the Philips TAMA data packets are in principle put through the µC and vice versa. Buffering, counting and checking of the data packets are only performed by the TAMA and the host in this case.
Mode select = '**3**' is the same as '**2**' with optional party line functionality.
With reader ID set to 0x00, no µC response packet buffering and host polling are necessary.
If the reader ID is set greater than 0x00 then party line is activated and host polling is mandatory.
(Party line details can be found in the polling command description ('aplxx')).

4.4.1 Mode select = '1' (high level language in binary format)

The intention for this mode is to use the checksum handling and the party line functionality for ASCII high level commands as well. All commands in High level language ASCII format (mode select = '0') are also available in binary format (mode select = '1'). The binary frame is build around the ASCII high level commands. Normally the binary format requires a software procedure for frame length and checksum calculation.

Binary frame from host to µC:

Mode select	Reader ID	Length	Data	Checksum
1 Byte	1 Byte	1 Byte	n Bytes	1 Byte

Mode select: Always '1' (0x31).
Reader ID: Unique ID of the selected reader.
Length: Number of Bytes of the user data.
Data: Length bytes of user data (high level command bytes)
Checksum: 0-(Summation of all bytes excluding mode select and checksum). (same as TAMA checksum algorithm).

Example for the high level Mifare® login command (l050E):
0x31 0x01 0x05 0x6C 0x30 0x35 0x30 0x45 0xB4
// mode, ID, Len, 'l050E', checkbyte

Binary frame from µC to host:

Mode select	Reader ID	Length	Data	Checksum
1 Byte	1 Byte	1 Byte	N Bytes	1 Byte

Mode select: Always '8' (0x38), indicates a packet from reader to host. (This is necessary for RS4xx half duplex mode to avoid that a reader interprets a response from another reader as a host command.)
Reader ID: Unique ID of the answering reader.
Length: Number of Bytes of the user data.
Data: Length bytes user data.
Checksum: 0-(Summation of all bytes excluding mode select and checksum). (Same as TAMA checksum algorithm).

'8' 0x01 0x08 0x46 0x46 0x30 0x30 0x30 0x30 0x30 0x30 0x4B
// mode, ID, Len, 'FF000000', checkbyte

Generally, if the received reader ID doesn't match to the stored ID inside the µC, then the µC sends for this command no response.

4.4.2 Mode select = '2' (TAMA language)

The Philips TAMA data packets are in principle put through the µC.

Mode select in ASCII	TAMA frame
1 Byte	n Bytes

Example:
Host to µC:
'2' 0x00 0x00 0xFF 0x02 0xFE 0xD4 0x02 0x2A 0x00 // GetFirmwareVersion (TAMA).

µC Response:

TAMA frame
n Bytes

0x00 0x00 0xFF 0x00 0xFF 0x00 // TAMA ACK packet
0x00 0x00 0xFF 0x04 0xFC 0xD5 0x03 0x02 0x02 0x24 0x00 // TAMA Version = 2.2

Refer to *PN531 User Manual* for more information about the TAMA packets.

4.4.3 Mode select = '3' (TAMA language) with addressingbyte for party line

Same as " mode select = '2' " except that two bytes are preceded.

Mode select in ASCII	Reader ID	TAMA frame
1 Byte	1 Byte	n Bytes

Party line example with reader address = 1:

Host to µC:
'3' 0x01 0x00 0x00 0xFF 0x02 0xFE 0xD4 0x02 0x2A 0x00 // GetFirmwareVersion (TAMA)
// of the Reader with the addressbyte 01.

µC Response:

Mode select in ASCII	Reader ID	TAMA frame
1 Byte	1 Byte	n Bytes

Mode select: Always '9' (0x39), indicates a packet from reader to host.
(This is necessary for RS4xx half duplex mode to avoid that a reader interprets a response from another reader as a host command.)

'9' 0x01 0x00 0x00 0xFF 0x00 0xFF 0x00 // TAMA ACK packet
'9' 0x01 0x00 0x00 0xFF 0x04 0xFC 0xD5 0x03 0x02 0x02 0x24 0x00
// TAMA Version = 2.2

Refer to *PN531 User Manual* for more information about the TAMA packets.

5. ARYGON high level language command set

Commands for adjustments and special µC-functions are always started with letter 'a'.

Common µC commands and µC adjustments

'ah'	Set UART baud rate to host side.
'at'	Set UART baud rate to TAMA side.
'ar'	Initiate a TAMA hardware reset.
'au'	Initiate a µC software reset.
'av'	Get µC firmware version.
'asn'	Get the unique serial number of the reader.
'as'	TAMA RFRegulation test.
'asl'	Power down mode (sleep).
'apc'	Configuration of the GPIO pins.
'apw'	Write GPIO.
'apr'	Read GPIO.
'apm'	Set PWM output duty cycle.
'apa'	Read analog port input value.
'ali'	Internal µC EEPROM login to modify EEPROM data.
'alo'	Internal µC EEPROM logout.
'aer'	Read one Byte from the internal µC EEPROM.
'aew'	Write one Byte into the internal µC EEPROM.
'aek'	Write a Mifare® login keytype with key into the internal µC EEPROM.
'aep'	Write a new pincode for internal µC EEPROM write access.
'apl'	Party line (RS4XX) polling command.

ISO 14443-A / Mifare commands

's'	Select a single card/tag
'l'	Login (authenticate)
'r'	Read data block/page on a tag
'wb'	Write data block/page (16 Byte)
'w4'	Write data block/page (4 Byte) for Mifare® ultralight tags
'rv'	Read value block
'wv'	Write/format value block
'+'	Increment value block
'-'	Decrement value block
'='	Copy value block
'h'	Set tag into halt
'of'	RF Configuration (switch off antenna)

5.1 µC response packet:

For every high level command the µC will send back at least one response packet.

Response packet format:

Packet start	Status/Error1	Status/Error2	User data length
2 character	2 character	2 character	2 character

Packet start: Always 'FF' (0x46 0x46), indicating a µC response packet.
Status/Error1: see error code list.
Status/Error2: see error code list.
User data length: Number of user data character in ascii-hex format.

Error Code List:

Error 1 Code	Error Cause
'00'	OK = no Error.
'01'	Ringbuffer, interrupt write overflow (TAMA side).
'02'	UART1 receiver framing or overrun error (TAMA side).
'03'	UART2 receiver framing or overrun error (host side).
'04'	TAMA receive packet checksum wrong (packet length or packet data).
'05'	Host receive packet checksum wrong (packet length or packet data).
'06'	Unknown mode-select command from host.
'07'	Ringbuffer, interrupt write overflow (host side).
'08'	One or more command parameter are out of range.
'09'	TAMA has detected an error at application level.
'0A'	No or wrong TAMA ack received after sending of TAMA command sequences.
'0B'	Wrong TAMA set baud rate response received after TAMA set baud rate command.
'0C'	Host or TAMA Command packet supervision Timer expired. Packet not complete within 1s.
'0D'	Ringbuffer write overflow (host side).
'0E'	No or wrong TAMA ack received after µC sends a command to TAMA.
'0F'	Host High level language checksum wrong.
'10'	Current block has no value block format (block format is corrupted).
'11'	Error during increment / decrement / copy value block. Error2 = TAMA Status after inc/dec. Refer to PN531 User Manual, error code list.
'12'	Baudrate not supported with the current low speed (low power) crystal.
'13'	Internal EEPROM read after write failed.
'14'	Internal EEPROM checksum failed (warning message).
'15'	Internal EEPROM address is out of the allowed range.
'16'	Internal EEPROM login missing. Application has no write access authorization.
'17'	Internal EEPROM login pincode wrong. Access denied.
'18'	Receive partylinebuffer overflow (from TAMA to µC).
'19'	Infomessage: No response available (partylinebuffer is empty).
'1A'	Optional LCD Busy check supervision Timer expired.

Error2 defines the sub Error of Error1.

Example:

µC Response packet	***Description***
'FF0C0000'	FF: µC response packet. 0C: Error1 = 0x0C, see error code list. 00: Error2 = 00, 0x0C has no sub error in this case. 00: no user data.
'FF0000044100'	FF: µC response packet. 00: Error1 = 0x00 = no error. 00: Error2 = 0x00 = no error. 04: 0x04 character user data following. '4100': user data.

5.2 Set UART baud rate to host side

This command is used to select the baud rate on the serial link between the host and the µC. The command response is sent back with the old baud rate. The µC switches to the new baud rate direct **after** sending of a successful response.
As default after power on the baud rate is always set to 9,6 kBaud.

Command	*Data*
'0ah'	Baud rate (1 Byte) 00 9.6 kBaud (default) 01 19.2 kBaud 02 38.4 kBaud 03 57.6 kBaud 04 115.2 kBaud 05 230.4 kBaud 06 460.8 kBaud (only with 5V µC version)

Response	*Description*
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (µC is now changing the baud rate to host side)

Example:

Command	*Description*
'0ah01'	Set Baud rate to 19.6 kBaud
'0ah06'	Set Baud rate to 460.8 kBaud for µC which runs with 5 Volt.

5.3 Set UART baud rate to TAMA side

To select the baud rate on the serial link between the µC and the TAMA.
The µC initiates automatically the necessary command sequence to adapt the TAMA baud rate.
For higher amount of data, peer to peer or T=CL data packets, it is mandatory to select the same baudrate on both sides of the µC (command '0ah' and '0at' with same parameter).
As default after power on the baud rate is always set to 9,6 kBaud.

Command	*Data*
'0at'	Baud rate (1 Byte) 00 9.6 kBaud (default) 01 19.2 kBaud 02 38.4 kBaud 03 57.6 kBaud 04 115.2 kBaud 05 230.4 kBaud 06 460.8 kBaud (only with 5V µC version)

Response	*Description*
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (µC will now change the baud rate to host side)

Example:

Command	*Description*
'0at01'	Set Baud rate to 19.6 kBaud
'0at06'	Set Baud rate to 460.8 kBaud for µC which runs with 5 Volt.

5.4 Initiate a TAMA hardware reset

It is possible to initiate a TAMA hardware reset with this command. The µC generates a 10ms pulse on the TAMA RSTPD Pin. After resetting the TAMA has a default UART baud rate of 9,6 kBaud. In this case the µC sets automatically its baud rate to TAMA side to 9,6 kBaud as well. It is important to know that the host application has to change the baud rate to the **old adjusted settings** before a TAMA reset was initiated (using command '0atxx') (The baud rate between µC and host remains unchanged).

Command	*Data*
'0ar'	None

Response	*Description*
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (TAMA has reset, baud rate between µC and TAMA is now 9,6 kBaud)

5.5 Initiate µC software reset

With this command the µC Software is reset and the bootloader code is executed. This command is very helpful if the host wants to initiate a µC firmware update without manipulating of any µC programmer pins (MCLR/VPP). Refer to *µC bootloader scenario*. During the µC reset the TAMA is reset as well.

The µC reset command requires a running firmware which is able to communicate with the host at least.

Command	*Data*
'0au'	None

Response	*Description*
'Bootloader987654321' 'ACMA'CRLF	Bootloader countdown. Normal firmware is running.

5.6 Get µC firmware version

This command displays the current running µC firmware variant and version.

Command	*Data*
'0av'	None

Response	*Description*
'FF00000600V0.1'CRLF	FF0000: Status, refer to µC response packet. 06: user data length (number of character in Hex). 00: µC firmware project variant. V0.1: firmware version, revision.

5.7 Get the unique serial number of the reader

During the first µC firmware programming, the µC manufacturer defines a 4 Byte long read only serial number. The host application has read access to this unique serial number.
In opposite to the "Reader ID for party line commands", the serial number can not be changed.

Command	*Data*
'0asn'	None

Response	*Description*
'FF00000813579BDF'CRLF	FF0000: Status, refer to µC response packet. 08: user data length (number of character in Hex). 13579BDF: unique serial number in Hex (= 0324508639 decimal).

5.8 TAMA RF regulation test

This command is used for easy handling of the TAMA radio regulation test command, refer to PN531 User Manual for details. TAMA starts an endless RF transmission.

Command	*Data*
'0as'	TxMode (1 Byte) , refer to PN531 User Manual for details

Response	*Description*
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (TAMA starts endless sending.)

Example:

Command	*Description*
'0as00'	RF Test with 106 kBaud and Mifare® modulation.
'0as12'	RF Test with 212 kBaud and FeliCa™ modulation.
'0as22'	RF Test with 424 kBaud and FeliCa™ modulation.

This command will never stop. The PN531 transmits data until a new command comes from the host controller.

5.9 Power down mode (Sleep)

A sleep mode is available for minimizing the µC and/or TAMA power consumption.
As first, TAMA has to be set into sleep mode and then the µC.

Two commands are available to set the TAMA into power down mode (as first, antenna must be **switched off** if it is not already done):
- PowerDown (Refer to PN531 User Manual for details):
- TgInitTAMATarget (Refer to PN531 User Manual for details):

(Direct TAMA commands are sent with µC mode select = '2')
After TAMA is asleep the µC can be set into sleep mode with the high level comand **'asl'**.

Wake up conditions

- External RF field is detected by TAMA during sleep (refer to TgInitTAMATarget).
  If the TAMA was set into sleep mode via command TgInitTAMATarget, then TAMA will automatically wake up the µC with wake up handshake. Then the TgInitTAMATarget response packet will automatically send through the µC to the host.
- Event occurred at SAM side (a SAM can be connected on the TAMA S2C interface).
- Falling edge detected on the following µC interrupt pin:
  RST/**UINT** pin from any external hardware circuit.
  H_REQ pin from the host (for wake up handshake from host side).
  IRQ from TAMA (for wake up handshake from TAMA side).
- Any Byte received on UART RX pin will cause an auto wake-up event, only if µC PIC18(L)F6621 is equipped. (PIC18F6520 does not support the UART auto wake up feature).

Wake up handshake

TAMA provides a wake up hardware handshake to warn the µC before sending a data packet. This is necessary to avoid losing of data bits because of no running µC UART periphery in sleep mode. (Refer to PN531 User Manual for details).
This wake up handshake is always implemented and configured between TAMA and µC.
A similar wake up handshake is provided between µC and host.

Host wants to wake up µC

The host asserts the H_REQ line to low and waits until the µC generates a low pulse on the IRQ pin with a pulse length of several µs. Now the µC is awaken and sends the dummy error packet '**FF060000**' to the host. This error packet can be ignored by the host and the next packet can be send to the µC. This next packet is normally a TAMA wake up command. The surest method to wake up TAMA is to perform a TAMA hardware reset via µC high level command **'ar'**. It is also possible to send any valid data packet to the TAMA to wake it up. The data are not interpreted, but this packet will be only used to wake up TAMA.
After wake up, the handshake mechanism is **no more needed**.

Only with PIC18F6621 or PIC18LF6621 equipped the host has the additional possibility to wake up the µC with a byte send to the µC UART Rx pin. No wake up handshake is needed in this case thanks to the enhanced UART of the PIC18(L)F6621.

To avoid data or framing errors the host has to send the wake up signal 0x00 to the µC.
The wake up signal (byte 0x00) must be followed by a sufficient interval to allow enough time for the µC oscillator to start and provide proper initialization of the UART.
About 10ms wake up time is enough for the worst case. (We have measured about 3ms).
After initialization of the UART, normally the packet 'FF060000' is sent to the host after wake up. But all Bytes or error packets receiving from µC can be ignored within the wake up time.

µC wants to wake up host

If µC wakes up, it generates always a low pulse on the IRQ pin with a pulse length in several µs. The host can use this trigger signal to wake up itself. Then normally the packet 'FF060000' is sent to the host which can be ignored or used as wake up event.
Now the host can send a packet to the µC.

µC sleep command:

Command	***Data***
'0asl'	None

Response	***Description***
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (µC is now in power down mode (sleep))

5.10 µC GPIO command set

The µC provides GPIO ports to give the user the possibility to build small hardware applications around the ACMA. The following I/O port pin types are available:

- 8 digital output or input pins.
- 1 open drain digital output or input pin.
- Up to 3 PWM output pins with a cycle duration (period) of T = 25µs (1/T = 40 kHz) and a duty cycle from 0% to 100 % with 1% resolution.
- 1 analog input pin with a voltage range of 0V … Vcc.

Port Pin assignment:

Logical pin number	***Physical hardware pin name***	***Description***
'00'	P0	GPIO
'01'	P1	GPIO
'02'	P2	GPIO
'03'	P3	GPIO
'04'	P4	GPIO
'05'	P5	GPIO
'06'	P6	GPIO
'07'	P7	GPIO
'08'	P8	GPIO (open drain characteristic)
PWM 0	PM0	PWM output
PWM 1	PM1	PWM output
PWM 2	PM2	PWM output
'associated with the command'	PA	Analog input for the ADC

5.10.1 GPIO pin configuration

The direction of the pins '00' … '08' can be configured. After power on the direction of the GPIO pins are set to input as default.

Command	*Data*
'0apc'	Port pin number (1 Byte) range: 00 … 08 (logical numbering). Port pin direction (1 Byte) 00 = output direction, 01 = input direction.

Response	*Description*
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (µC has received the command, GPIO pin configuration is successful).

Example:

Command	*Description*
'0apc0100'	Set the direction of the logical pin 01 to output.
'0apc0601'	Set the direction of the logical pin 06 to input.

5.10.2 GPIO pin write

If the port pin direction is correctly set to output, then with this command the pin can be driven to high (logical 01) or to low (logical 00).
Please note that port pin 08 has an open drain characteristic.

Command	*Data*
'0apw'	Port pin number (1 Byte) range: 00 … 08 (logical numbering). Port pin value (1 Byte) 00 = low, 01 = high.

Response	*Description*
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (µC has received the command, GPIO pin is set).

Example:

Command	*Description*
'0apw0200'	Set the logical pin 02 to low.
'0apw0501'	Set the logical pin 05 to high.

5.10.3 GPIO pin read

If the port pin direction is correct set to input, then with this command the external pin status can be read.

Command	Data
'0apr'	Port pin number (1 Byte) range: 00 … 08 (logical numbering).

Response	Description
'FF000002xx'CRLF	FF0000: Status, refer to µC response packet. 02: user data length (number of character in Hex). xx: Status (value) of the pin.

Example:

Command	Description
'0apr03'	Read the logical pin 02.

Response	Description
'FF00000201'CRLF	FF0000: Status, refer to µC response packet. 02: user data length (number of character in Hex). 01: The port pin 03 is high (logical 01)

or

Response	Description
'FF00000200'CRLF	FF1600: Status, refer to µC response packet. 02: user data length (number of character in Hex). 01: The port pin 03 is low (logical 00).

5.10.4 GPIO PWM, set duty cycle

The µC provides some PWM output pins for user applications.
Example given:
- This pins, connected via a lowpass filter, can be used to generate a variable analog output voltage (D/A converter).
- A relay or a photoelectric relay can be energized with an adjustable reduced current.
- …

The PWM output pins have all a fixed cycle duration of T = 25µs (1/T = 40 kHz) and a variable duty cycle range from 0% (0x00) to 100 % (0x64) with 1% resolution.

To reduce the µC power consumption the user (host) has the possibility to switch off the µC PWM periphery. This can be done by setting the PWM duty cycle of any PWM port to the special value 0xff.

Note:
After switching off the PWM periphery, the static value (high or low) at all PWM port pins can be undefined. It depends on the last level when the switch off command is sent and therefore the port pin level is random.

To avoid this the user (host) should set the PWM duty cycle to 0% or to 100% for each PWM port before. After these settings the user can be sure that the desired level is statically fixed when the PWM is stopped.
With any other valid value for any PWM Port between 0x00 … 0x64 the PWM periphery is automatically switched on.

Command	***Data***
'0apm'	PWM pin number (1 Byte) range: 00 … 02 (logical numbering). (for number and locations of the pins, please refer to our ANHW-Axxxx-xx.pdf) Duty Cycle (1 Byte) range: 00 … 64 (equivalent to 0% … 100%). special: ff = switch PWM periphery off for power consumption reduction.

Response	***Description***
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (µC has received the command, PWM is set).

Example:

Command	***Description***
'0apm001A'	PWM duty cycle of the PWM0 is set to 26%.
'0apm0100'	PWM duty cycle of the PWM1 is set to 00% (pin is permanently set to low).
'0apm0264'	PWM duty cycle of the PWM2 is set to 100% (pin is permanently set to high).
'0apm01ff'	All PWMs are switched off (pin level depends on the last level of each PWM pin).

5.10.5 GPIO Analog input pin

Analog voltages from 0V up to Vcc can be read and converted into a 10 Bit digital value (0x0000 up to 0x03FF). That means we have an analog to digital converter with a 10 bit resolution.

Command	***Data***
'0apa'	Port pin number (1 Byte) port: 01 (logical analog port number. At the moment only port 01 is available).

Response	***Description***
'FF000004xxyy'CRLF	FF0000: Status, refer to µC response packet. 04: user data length (number of character in Hex). xx: Highbyte of the 10 bit analog to digital conversion. yy: Lowbyte of the 10 bit analog to digital conversion.

Example:

Command	***Description***
'0apa01'	Read the analog input port with logical port number 01.

Response	***Description***
'FF0000040294'CRLF	FF0000: Status, Refer to µC response packet. 04: user data length (number of character in Hex). 02: Highbyte of the 10 bit analog to digital conversion. 94: Lowbyte of the 10 bit analog to digital conversion. Example for µC Vcc = 5V: 0x03FF (1023) => 5V 0x0294 (660) => **3,23V**

5.11 µC EEPROM commands

The µC is able to store non-volatile data in its internal EEPROM. The host application can store their flags, the reader ID, Mifare® keys and so on to customize the reader behavior.

The personalization of the reader, Mifare® key storing into the µC EEPROM and call them just by a key number later, is an important utility against malicious attacks.

Because of the limited erase/write cycle (typically 1000 000), the µC firmware performs always a read after write verify. If the host receives an error free µC write response, then the user can be sure that the last write access was successfully verified. A read after write check by the host application is not necessary.
Additional after every power on, the complete EEPROM is compared with a stored checksum byte. An error will be sent in case of failure.

EEPROM memory organization (EEPROM size = 1024 Byte)

EEPROM Register	***Description***
0x0000	rfu.
0x0001 … 0x0004	EEPROM login pincode.
0x0005	Unique Reader ID for party line commands. **Default ID = 0x01.**
0x0006	Stand-alone On (0x01) / Off (0x00). **Default is stand-alone off (0x00)**.
0x0007 … 0x0009	rfu.
0x000A	Mifare® key type for Mifare® authentication (key number 0x00).
0x000B … 0x0010	Mifare® key (6 Byte) for Mifare® authentication (key number 0x00).
0x0011	Mifare® key type for Mifare® authentication (key number 0x01).
0x0012 … 0x0017	Mifare® key (6 Byte) for Mifare® authentication (key number 0x01).
…	…
0x00E3	Mifare® key type for Mifare® authentication (key number 0x1F).
0x00E4 … 0x00E9	Mifare® key (6 Byte) for Mifare® authentication (key number 0x1F).
0x00EA … 0x03FD	User free read/write data area.
0x03FE	EEPROM data valid flag.
0x03FF	EEPROM checksum Byte.

5.11.1 µC EEPROM write access (login and logout command)

EEPROM login:
For safety reasons an EEPROM data modify is only possible after a successful login command with a correct pincode as parameter.
The transport pincode is set to **'ff' 'ff' 'ff' ff'** as default.

Copy Command	***Data***
'0ali'	Pincode (4 Byte) every pincode Byte can have values from 00 to ff.

Response	***Description***
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (µC has received the command, EEPROM login is successfully).

Example:

Command	*Description*
'0ali0a04cdfe'	EEPROM login with pincode 0x0a, 0x04, 0xcd, 0xfe.
'0aliffffffff'	EEPROM login with default transport pincode.

Response	*Description*
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (µC has received the command, EEPROM login is successfully).

or

Response	*Description*
'FF170000'CRLF	FF1700: Status, refer to µC response packet. 17: Pincode wrong, EEPROM access denied.

EEPROM Logout:

After all EEPROM modifications are done, **do not forget to logout** !
After power on EEPROM logout is default.

Copy Command	*Data*
'0alo'	None

Response	*Description*
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (µC has received the command, EEPROM logout is successfully).

5.11.2 µC EEPROM write command

The host application has write access only after a successful EEPROM login command.
All Bytes are writable except:
- The EEPROM login pincode.
- The stored Mifare® keys with key types.

(To modify the pincode or the Mifare® keys, special commands exits, see later).
A read after write verify and an EEPROM checksum update are automatically performed by this command.

Copy Command	***Data***
'0aew'	EEPROM address (4 Byte) EEPROM address range: 0000 … 03FF (1024 Bytes). EEPROM data to write (1 Byte) valid data: 00 … FF

Response	***Description***
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (µC has received the command, EEPROM write was successful).

Example:

Command	***Description***
'0aew00eaAB'	Write the Byte 0xAB to the EEPROM address 0x00EA.

Response	***Description***
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (µC has received the command, EEPROM write was successful).

or

Response	***Description***
'FF160000'CRLF	FF1600: Status, refer to µC response packet. 16: EEPROM login missing, EEPROM write access denied.

or

Response	***Description***
'FF130000'CRLF	FF1300: Status, refer to µC response packet. 13: EEPROM Read-after-write failed.

5.11.3 µC EEPROM read command

Stored EEPROM Bytes can be read via this command. For secure reasons the following data areas are not accessible because they are write only:
- The EEPROM login pincode.
- The stored Mifare® Keys with key types.

In opposition to the write command, the read command requires **no** EEPROM login.

Copy Command	***Data***
'0aer'	EEPROM address (4 Byte) EEPROM address range: 0000 … 03FF (1024 Bytes).

Response	***Description***
'FF000002xx'CRLF	FF0000: Status, refer to µC response packet. 02: user data length (number of character in Hex). xx: µC EEPROM read result.

Example:

Command	***Description***
'0aer00ea'	Read the contents of the EEPROM address 0x00EA.

Response1	***Description***
'FF000002AB'CRLF	02: user data length (number of character in Hex). AB: contents of the EEPROM address 0x00EA.

Additionally the EEPROM checksum is verified during a read command, because of the limited EEPROM erase/write cycle. Only in case of a not successful check, the host is informed with the following warning message:

Response2 (optional)	***Description***
'FF140000'CRLF	FF1400: Status, refer to µC response packet. 14: EEPROM checksum failed.

5.11.4 µC EEPROM saves Mifare® authentication key

For personalization of the reader, the host can store up to 32 Mifare® keys with key type into the µC EEPROM. After personalization the host can use the high level Mifare® login command to use the internal stored keys (refer to the high level login command '0lxxxx') .

The "save Mifare® authentication key command" requires a successfully EEPROM login.

Copy Command	***Data***
'0aek'	EEPROM 'key number / storage number' (1 Byte) EEPROM 'key number'. Valid range 00 … 1F. (The physical EEPROM address is calculated by the µC.) Mifare® key type (1 character) 'A' Authentication using key type A 'B' Authentication using key type B Mifare® key (6 Bytes) e.g.: 112233aabbcc

Response	***Description***
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (µC has received the command, EEPROM write is successful).

Example:

Command	***Description***
'0aek0CB2ff8877a0c1d'	Write into the EEPROM at key number (storage number) 0x0C the Mifare® key type 'B' and the Mifare® key '2ff8877a0c1d'.

Response	***Description***
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (µC has received the command, EEPROM write is successfully).

or

Response	***Description***
'FF160000'CRLF	FF1600: Status, refer to µC response packet. 16: EEPROM login missing, EEPROM write access denied.

or

Response	***Description***
'FF130000'CRLF	FF1300: Status, refer to µC response packet. 13: EEPROM read after write failed.

5.11.5 µC EEPROM modify login pincode

For safety reasons and unintentionally EEPROM write access an EEPROM modify is only possible after a successful login command with a correct pincode as parameter. The transport pincode is set to **'ff' 'ff' 'ff' ff'** as default.
After login the user can change the pincode to any other value.
The user has to handle **carefully** if he changes the pincode !
If the code is lost, only ARYGON can help to get again access to the EEPROM.

The "modify pincode command" requires a successfully EEPROM login.

Copy Command	***Data***
'0aep'	New pincode (4 Byte) every pincode Byte can have values from 00 to ff.

Response	***Description***
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (µC has received the command, EEPROM write is successfully).

Example:

Command	***Description***
'0aep01020a0b'	Set the new pincode to 0x01 0x02 0x0a 0x0b.

Response	***Description***
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (µC has received the command, EEPROM write is successful).

or

Response	***Description***
'FF160000'CRLF	FF1600: Status, refer to µC response packet. 16: EEPROM login missing, EEPROM write access denied.

or

Response	***Description***
'FF130000'CRLF	FF1300: Status, refer to µC response packet. 13: EEPROM read after write failed.

5.12 Party line (RS4XX) polling command

In party line environment, every reader acts as a pure slave which is connected to one master (host).
A reader is only allowed to answer after a polling request command from the host to avoid RS4XX bus collisions.

The party line mode is activated if:
- All readers, which are connected to a RS4XX bus, have a different reader ID greater than 0x00. (refer to µC EEPROM commands for changing of the reader ID.)
- All host commands must be sent with mode select = '1' or '3'.

In party line environment it is **not allowed** to send commands with mode select = '0' or '2'.

Mode select = '0'. (no party line)

Command	*Data*
'0apl'	µC or TAMA response (1 Byte) 01 Polling for µC high level response packet (Mode 1). 03 Polling for TAMA low level response packet (Mode 3).

Response	*Description*
'FF190000'CRLF	FF0000: Status, refer to µC response packet. (In this case '19' is no error. It is only an info that a response packet is not yet available).

or

Response	*Description*
'FF0000yyxxzz…'	FF0000: Status, refer to µC response packet. Any µC high level response packet.

It makes only sense to use the polling command **in party line mode** of course. Only for completeness the polling command works in Mode 0 without party line as well.

Mode select = '1' (party line). apl in binary format ('1' 0x01 0x05 'apl' '01' checks).

Command	*Data*
31 01 05 61 70 6C 30 31 5C	µC or TAMA response (1 Byte) 01 Polling for µC high level response packet (Mode 1).

Response for mode select = '1'. ('8' 0x01 0x08 'FF190000' checks).

Response	*Description*
38 01 08 46 46 31 39 30 30 30 30 41	0x38: fixed to '8' 0x01: Reader ID 0x08: packet length (without mode select, reader ID, checksum) 'FF1900': Status, refer to µC response packet. (In this case '19' is no error. It is only an info that a response packet is not yet available). 0x00: user data length is 0.

or

If any data packet is buffered and available in the µC, then always one µC packet is sent to the host.
For example, µC get Firmware Version Response packet: ('8' 0x01 0x0E 'FF00000600V0.6' checks).

Response	*Description*
38 01 0E 46 46 30 30 30 30 30 36 30 30 56 30 2E 36 F5	0x38: fixed to '8' = Response for mode select '1' command. 0x01: Reader ID 0x0E: packet length (without mode select, reader ID, checksum) FF0000: Status, refer to µC response packet. 06: user data length (number of character in Hex). 00: µC Firmware project variant. V0.6: Firmware version, revision.

Mode select = '3' (party line). apl in binary format ('1' 0x01 0x05 'apl' '03' checks).

Command	***Data***
31 01 05 61 70 6C 30 33 5A	µC or TAMA response (1 Byte) 03 Polling for TAMA low level response packet (Mode 3). (The command itself is still sent with mode '1', because it is a µC command and no TAMA packet.)

Response for Mode select = '3'. ('8' 0x01 0x08 'FF190000' checks).

Response	***Description***
38 01 08 46 46 31 39 30 30 30 30 41	0x38: fixed to '8' = Response for mode select '1' command. 0x01: Reader ID 0x08: packet length (without mode select, reader ID, checksum) 'FF1900': Status, Refer to µC response packet. (In this case '19' is no error. It is only an info that a response packet is not yet available). 0x00: user data length is 0. (If no TAMA response is available, the response itself is still sent with mode 8')

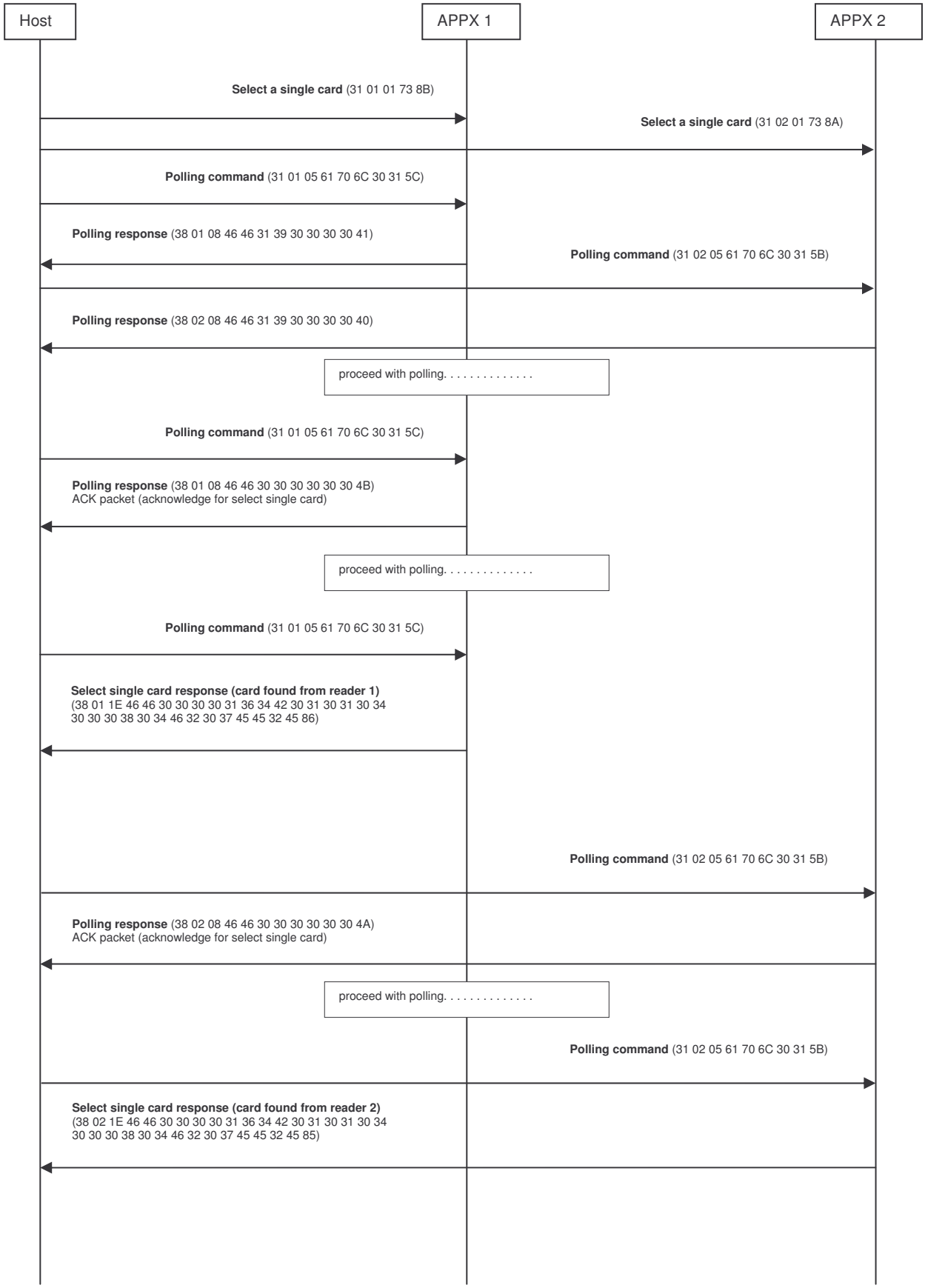
or

If any packet is stored and available in the µC, then always one TAMA packet is sent to the host.
For example TAMA Ack packet: ('9' 0x01 TAMA packet).

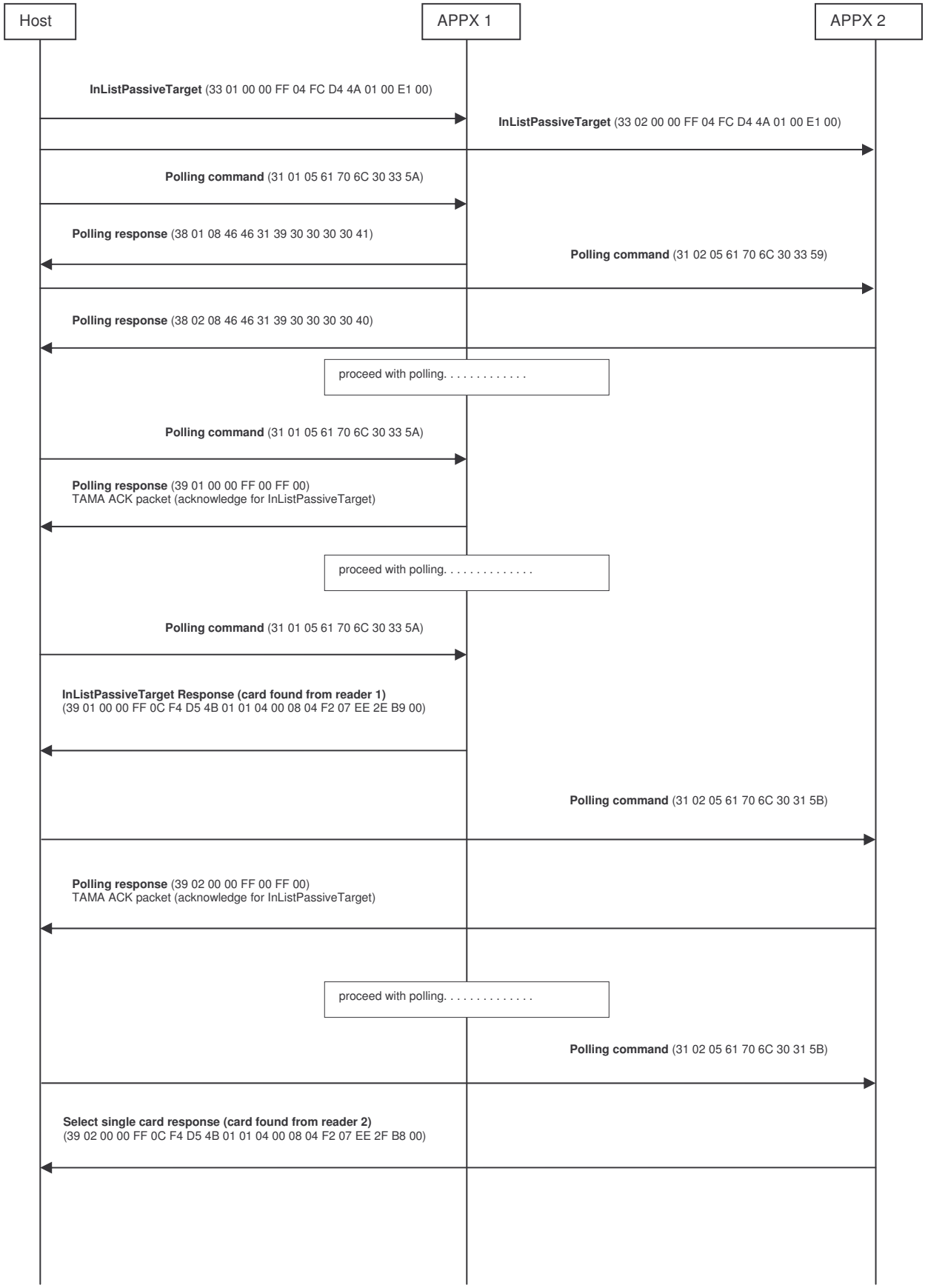
Response	***Description***
39 01 00 00 FF 00 FF 00	0x39: fixed to '9' = Response for mode select '3' command. 0x01: Reader ID 00 00 FF 00 FF 00: TAMA Ack packet

Following typical party line scenarios are shown in mode select '1' and '3'. In this example two APPx modules are connected via a RS4XX bus to one host.

Party line RS4XX example in high level language (mode select = '1').



Party line RS4XX example in TAMA language (mode select = '3').



5.13 Select a single card

Select a single 106 kBaud target in the antenna field. In case of success the command returns the tag ID (NFCID1) and the type of the selected target (tag).

Command	*Data*
'0s'	None

Response1	*Description*
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (µC and TAMA has received the command and TAMA starts scanning).

Target found (e.g.: Mifare® standard target):

Response2	*Description*
'FF0000164B01010400080432EEED2E'CRLF	FF0000: Status, refer to µC response packet. 16: User data length (number of character in Hex). 4B: TAMA InListPassiveTarget Response packet. 01: Number of initialized targets. 01: Target Number 0400: SENS_RES 08: SEL_RES (card Type) (08 = Mifare® standard) 04: Card ID (NFCID1) length. 32EEED2E: Card ID (NFCID1). (Refer to PN531 User Manual)

Target found (e.g.: Mifare® standard 4kByte target):

Response2	*Description*
'FF0000164B0101020018045634740 0'CRLF	FF0000: Status, refer to µC response packet. 16: User data length (number of character in Hex). 4B: TAMA InListPassiveTarget response packet. 01: Number of initialized targets. 01: Target number 0200: SENS_RES 18: SEL_RES (card Typ) (18 = Mifare® 4k) 04: Card ID (NFCID1) length. 56347400: Card ID (NFCID1).

Target found (e.g.: Mifare® Ultralight target):

Response2	*Description*
'FF00001E4B0101440000088804686211127A00'CRLF	FF0000: Status, refer to µC response packet. 1E: User data length (number of character in Hex). 4B: TAMA InListPassiveTarget response packet. 01: Number of initialized targets. 01: Target number 4400: SENS_RES 00: SEL_RES (card Typ) (00 = Mifare® UltraLight) 08: Card ID (NFCID1) length. 8804686211127A00: Card ID (NFCID1).

Remark:
The TAMA is able to handle two cards in the field at the same time without taking care of selecting/deselecting of the other card. This is possible in TAMA language.
- InListPassiveTarget (0x4A 0x02 0x00) (Refer to PN531 User Manual).
- InDataExchange (0x40 Target number …) (Refer to PN531 User Manual).

With the High level command '0s' the cards are only alternately selected.

5.14 Mifare® login (authenticate)

It performs an authentication to access one sector of a card. *Only* one sector can be accessed at the same time. The authentication key can be transmitted directly from the host or for safety reasons indirectly via stored keys and key type from the µC-EEPROM to the TAMA.
To login into a complete sector **any** blockaddress belonging to this sector can be chosen as command parameter.
The login requires a successful select.

Command	***Data***
'0l'	Block address (1 Byte) 00 or 01 or 02 or 03 = sector 0 04 or 05 or 06 or 07 = sector 1 ... (Refer to Mifare® card documentation for memory organization details.) Key control Byte (1 Byte) 00 … 1F authenticate with stored key type A or B and key 0x00 … 0x1F. FF authenticate with key given with this command (at last parameter). Key type (1 character) optional 'A' Authentication using key type A 'B' Authentication using key type B Key (6 Bytes) optional 112233445566

Response1	***Description***
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (µC and TAMA has received the command and TAMA starts login).

Response2	***Description***
'FF0000044100'CRLF	FF0000: Status, refer to µC response packet. 04: User data length (number of character in Hex). 41: TAMA InDataExchange Response packet. 00: TAMA status, refer to PN531 User Manual, error code list.

Example:

Command	***Description***
'0l21FFAaabbccddeeff'	Authenticate for block 0x21 (= sector 8) with key type A using key aabbccddeeff.
'0l050E'	Authenticate for block 0x05 (= sector 1) using stored key 0x0E from the µC EEPROM including stored key type.
'0l0DFFBfffffffffff'	Authenticate for block 0x0D (= sector 3) with key type B using Philips transport Key ffffffffffff.

5.15 Read data block (page)

This command reads a data block (page) on a target. The size of returned valid data depends on the used tag. The block address range depends on the present tag as well. The reading requires a successful login in case of Mifare® tags with authentication.

Command	***Data***
'0r'	Block address / page address (1 Byte)

Response1	***Description***
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (µC and TAMA has received the command and TAMA starts reading).

Block/page read (e.g.: Mifare® standard target)

Response2	***Description***
'FF0000244100010203040506070809 0A0B0C0D0E0F10'	FF0000: Status, refer to µC response packet. 24: User data length (number of character in Hex). 41: TAMA InDataExchange response packet. 00: TAMA tatus, refer to PN531 User Manual, error code list. 01…11: 16 Bytes block data. (Refer to Mifare® card documentation.)

Block/page read (e.g.: Mifare® Ultralight target)

Response2	***Description***
'FF0000244100010203040506070809 0A0B0C0D0E0F10'	FF0000: Status, refer to µC response packet. 24: user data length (number of character in Hex). 41: TAMA InDataExchange Response packet. 00: TAMA Status, refer to PN531 User Manual, error code list. 01…04: 4 Byte data from page x 05…08: 4 Byte data from page x+1 09…0C: 4 Byte data from page x+2 0D…10: 4 Byte data from page x+3 (Refer to to Mifare® ultralight card documentation.)

Example:

Command	***Description***
'0r21'	Read the data of block 0x21.

5.16 Write data block (16 Byte)

This command writes data to a block/page. The writing requires a successful login in case of Mifare® tags with authentication.

Command	*Data*
'0wb'	Block/page address (1 Byte) Data Bytes (16 Byte)

Response1	*Description*
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (µC and TAMA has received the command and TAMA starts writing).

Response2	*Description*
'FF0000044100'CRLF	FF0000: Status, refer to µC response packet. 04: User data length (number of character in Hex). 41: TAMA InDataExchange Response packet. 00: TAMA Status, refer to PN531 User Manual, error code list.

Example:

Command	*Description*
'0wb2101020304050607080 90A0B0C0D0E0F10'	Mifare® Standard target: Write data on block 0x21 (= sector 8).
'0wb0401020304050607080 90A0B0C0D0E0F10'	Mifare® Ultralight target (compatibility write). Write data 01020304 on page 0x04. Data Bytes 4…15 are ignored, refer to Mifare® ultralight card documentation.)

5.17 Write data block (4 Byte)

This command writes 4 Byte data to a Mifare® ultralight page. It can be used for programming of the Mifare® ultralight OTP (One Time Programming) pages as well.

Command	*Data*
'0w4'	page address (1 Byte) Data Bytes (4 Byte)

Response1	*Description*
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (µC and TAMA has received the command and TAMA starts writing).

Response2	*Description*
'FF0000044100'CRLF	FF0000: Status, refer to µC response packet. 04: user data length (number of character in Hex). 41: TAMA InDataExchange response packet. 00: TAMA Status, refer to PN531 User Manual, error code list.

Example

Command	*Description*
'0w40401020304'	Write data 01020304 on page 0x04, refer to Mifare® ultralight card documentation.

5.18 Read value block

This command reads the value from a value block on a target. A check is performed if the data is in value block format.
The reading value block requires a successful login, a correct formatted value block as source and correct settings of the access condition bits (refer to Mifare® card documentation).

Command	***Data***
'0rv'	Block address (1 Byte)

Response1	***Description***
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (µC and TAMA has received the command and TAMA starts reading).

Response2	***Description***
'FF00000C410000000104'	FF0000: Status, refer to µC response packet. 0C: User data length (number of character in Hex). 41: TAMA InDataExchange response packet. 00: TAMA Status, refer to PN531 User Manual, error code list. 00000104: 4 Byte value block data.

Example:

Command	***Description***
'0rv21'	Read the value of value block 0x21.

Responses	***Description***
'FF000000'CRLF 'FF00000C410000000104'	FF0000: Status, refer to µC response packet. The value block 21 has the value 0x00000104. No error detected.

or

Responses	***Description***
'FF000000'CRLF 'FF100000'	FF0000: Status, refer to µC response packet. µC Error 0x10. Block 21 has no value block format, refer to µC response packet.

or

Responses	***Description***
'FF000000'CRLF 'FF0000044114'	FF0000: Status, refer to µC response packet. TAMA Error 0x14 authentication error, refer to PN531 User Manual, error code list. The block 0x21 is outside of the current authenticated sector.

According to the Mifare® card documentation the value Byte are stored with LSB first, once inverted and twice non inverted and four byte for the address. That means only 4 Byte are available for the value. The read value block command will reverse all value Byte for the response packet to the host.

Example:
A Mifare® card has stored in value block 0x21 260 units.
260 units = 0x104 = 0x00 0x00 0x01 0x04.

Inside the card memory we found:
04 01 00 00 FB FE FF FF 04 01 00 00 00 FF 00 FF

In case of no error the following read value block response packet is sent:
…00 00 01 04. // value is Byte reversed.

5.19 Write value block

This command formats a block as a value block containing a 32-Bit value. Value blocks need a complete 16-byte block due to redundant storage.

The writing value block requires a successful login and correct settings of the access condition bits, refer to Mifare® card documentation.

Command	***Data***
'0wv'	Block address (1 Byte) Value (4 Byte)

Response1	***Description***
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (µC and TAMA has received the command and TAMA starts writing).

Response2	***Description***
'FF0000044100'	FF0000: Status, refer to µC response packet. 04: User data length (number of character in Hex). 41: TAMA InDataExchange response packet. 00: TAMA status, refer to PN531 User Manual, error code list.

Example:

Command	***Description***
'0wv2100000005'	Writes the value 0x00000005 to block 0x21.

According to the Mifare® card documentation the value Byte are stored with LSB first, once inverted and twice none inverted and four byte for the address. That means only 4 Byte are available for the value. The write value block command will reverse all value Byte before sending to the TAMA.

Example:
A user wants to store 430 units into block 0x20 of the Mifare® card.
430 units = 0x01AE = 0x00 0x00 0x01 0xAE.

After sending of '0wv20000001AE', we found in block 0x20 inside the card memory:
AE 01 00 00 51 FE FF FF AE 01 00 00 00 FF 00 FF

The write value block command is designed to create blocks, which match the value format. This command requires write access to specified block. It is not recommended to use this instruction for ticketing operation. For ticketing application special instructions (increment/decrement/copy) are supported.

5.20 Increment/decrement value block

Increments or decrements a value block with a defined value.
These commands require a successful login, a correct formatted value block as source and correct settings of the access condition bits (refer to Mifare® card documentation).

Increment Command	***Data***
'0+'	Block address (1 Byte) Value (4 Byte)

Decrement Command	***Data***
'0-'	Block address (1 Byte) Value (4 Byte)

Response1	***Description***
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (µC and TAMA has received the command and TAMA starts inc/dec and transfer).

Response2	***Description***
'FF0000044100'	FF0000: Status, refer to µC response packet. 04: User data length (number of character in Hex). 41: TAMA InDataExchange response packet. 00: TAMA status, refer to PN531 User Manual, error code list.

Example:

Command	***Description***
'0+2100000005'	Adds 5 to value block 0x21.
'0-210000010F'	Subtract 271 to value block 0x21.

Responses	***Description***
'FF000000'CRLF 'FF0000044100'	FF0000: Status, refer to µC response packet. OK, no error detected.

or

Response	***Description***
'FF110100'CRLF	FF1101: Status, refer to µC response packet. 11: Error detected during inc/dec value block. 01: TAMA has detected the Error 0x01 during the inc/dec command. (Refer to PN531 User Manual, error code list). Transfer is canceled.

or

Responses	***Description***
'FF000000'CRLF 'FF0000044101'	FF0000: Status, refer to µC response packet. Inc/dec was successful executed but the transfer command fails. Possible reason: The current block is not in value format, refer to PN531 User Manual, error code list.

The µC generates two Mifare® card commands in sequence for the inc/dec command. First the *inc/dec card command* followed by the *transfer card command*.
Depending where the error occurs we see another error reporting from µC (see above).

The inc/dec commands will reverse all value Byte before sending to the TAMA.

5.21 Copy value block

Copies a value block to another block of the same sector. Used for backup and error recovery.
This command requires a successful login, a correct formatted value block as source and correct settings of the access condition bits (refer to Mifare® card documentation).

Copy Command	*Data*
'0='	Source block (1 Byte) Target block (1 Byte)

Response1	*Description*
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (µC and TAMA has received the command and TAMA starts copying).

Response2	*Description*
'FF0000044100'	FF0000: Status, refer to µC response packet. 04: User data length (number of character in Hex). 41: TAMA InDataExchange response packet. 00: TAMA status, refer to PN531 User Manual, error code list.

Example:

Command	*Description*
'0=2122'	Copy value block 21 to block 22.

Responses	*Description*
'FF000000'CRLF 'FF0000044100'	FF0000: Status, refer to µC response packet. OK, no error detected.

or

Response	*Description*
'FF110100'CRLF	FF1101: Status, refer to µC response packet. 11: Error detected during copying value block. 01: TAMA has detected the error 0x01 during the copy command, refer to PN531 User Manual, error code list. Transfer is canceled.

or

Responses	*Description*
'FF000000'CRLF 'FF0000044101'	FF0000: Status, refer to µC response packet. Block restore was successful executed but the transfer command fails, refer to PN531 User Manual, error code list.

The µC generates two Mifare® card commands in sequence for the copy command. First the *restore card command* followed by the *transfer card command*.
Depending where the error occurs we see another error reporting from µC (see above).

5.22 Set tag into halt

This command sets a selected or all tags in the field into halt state.

Copy Command	*Data*
'0h'	Tag ID, logical number of the relevant target (1 Byte) 00 = all available targets in the field. (01 = first target which was initialized.) (02 = second target which was initialized.)

Response	*Description*
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (µC and TAMA has received the command and TAMA sends the halt command).

Response2	*Description*
'FF0000044500'	FF0000: Status, refer to µC response packet. 04: User data length (number of character in Hex). 45: TAMA InDeselect response packet. 00: TAMA Status, refer to PN531 User Manual, error code list.

Example:

Command	*Description*
'0h00'	Set all selected targets in the field into halt state.

Remark:

Because of the fact that with the high level language only one tag can be selected at the same time, it makes no sense to use the halt command with other parameters than with 00. ('0h00') should be always used.

5.23 RF configuration (switch off antenna)

This command can be used to switch off the **RF** field. The antenna is automatically switched on during a select command.

RFCA (RF field Collision avoidance)
When **off**, the TAMA does not need to take care of external field before switching on its own field. The TAMA will generate RF field whatever external field is (present or not).

Copy command	***Data***
'0of'	RF configuration (1 Byte) 00 = RFCA off, RF off 01 = RFCA off, RF on 02 = RFCA on, RF off 03 = RFCA on, RF on

Response	***Description***
'FF000000'CRLF	FF0000: Status, refer to µC response packet. (µC and TAMA has received the command).

Response2	***Description***
'FF00000233'	FF0000: Status, refer to µC response packet. 02: User data length (number of character in Hex). 33: TAMA RFConfiguration response packet, refer to PN531 User Manual for more details.

Example:

Command	***Description***
'0of02'	Switch antenna off. RFCA is on.
'0of00'	Switch antenna off. RFCA is off.

6. Scenario examples in TAMA language

To transfer larger amount of data is the intention of NFC peer to peer and the T=CL protocol. At the moment it makes no sense to create a high level language in ASCII format for this.

A simplification or reduction of the transferred data is not possible. The TAMA language provides already a simplified language which can be used without having deep knowledge in the ECMA 340 standard for example.

With µC equipped, mode select = '2' or '3' (TAMA language) have to be used as interface protocols.

The TAMA commands for NFC peer to peer and T=CL are described in the Philips PN531 User Manual *UM0501-xx.pdf.* ARYGON will give support to start faster with this new protocol.

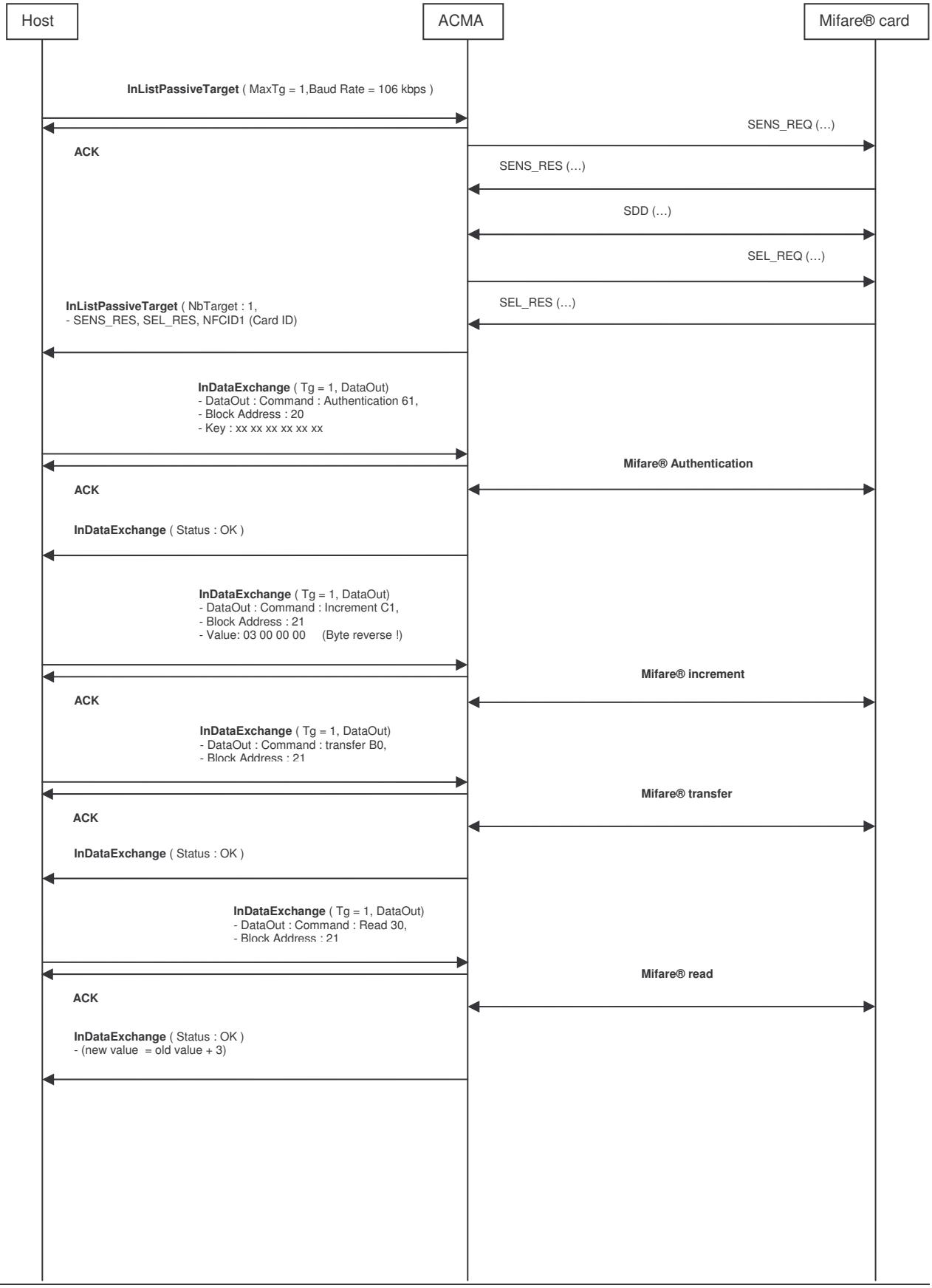
For this ARYGON has created a script based PC Tool named 'Reader Tester' which provides many peer to peer, T=CL, Desfire and Mifare® examples (refer to our Demokit documentation getting_started.pdf).

How the TAMA protocol works, the best explanation are shown in scenarios. Some important scenarios can be found in this document. For more scenarios and details please refer to User Manual *UM0501-xx.pdf.*

6.1 ISO14443-3 / Mifare® scenario in TAMA language:

Mifare® card reading/writing is shown in User Manual *UM0501-xx.pdf.*
In addition to the Philips User Manual, following is an increment, transfer and read scenarios shown. This is done in TAMA language (with µC mode select = '**2**' or no µC equipped) for Mifare® standard tags. The following command sequence requires a correct formatted value block as source and correct settings of the access condition bits (refer to Mifare® card documentation).

Mifare® value block increment, transfer and read example.



6.2 Other ISO14443-3 tags using the TAMA "InCommunicateThru" command

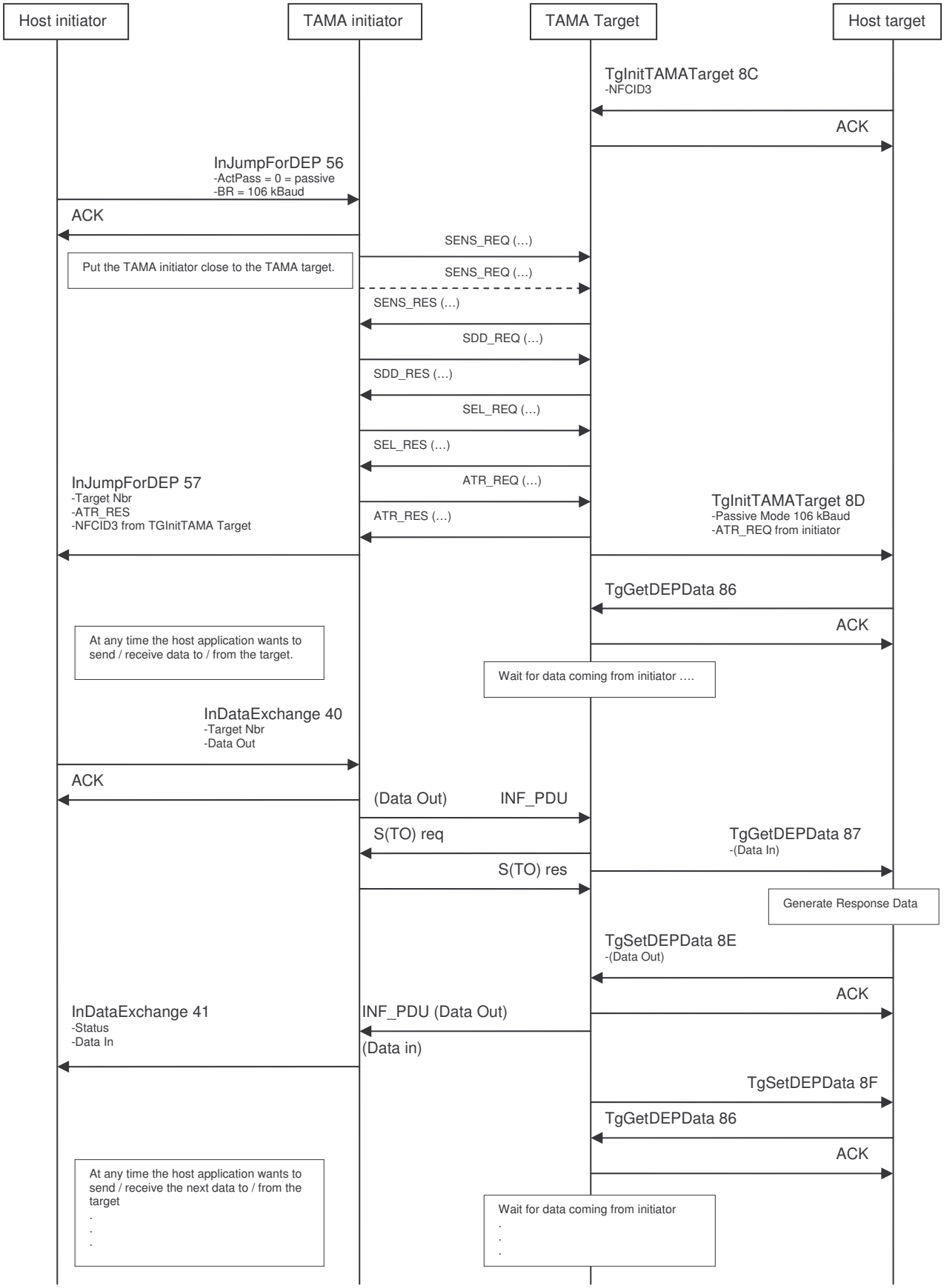
Scenario Planned.

6.3 ISO14443-4 (T=CL) scenarios:

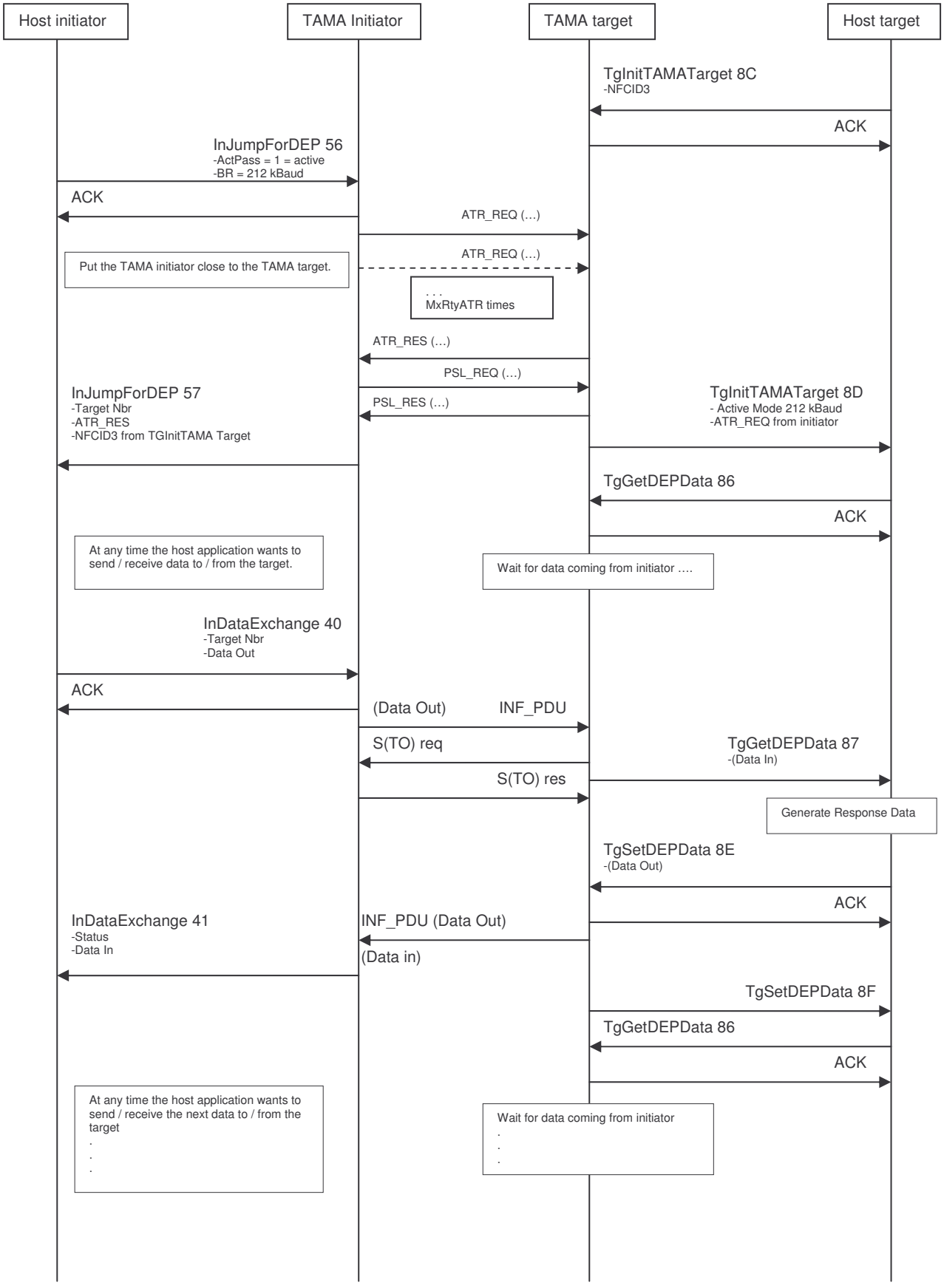
Scenario Planned.

6.4 NFC peer to peer scenarios:

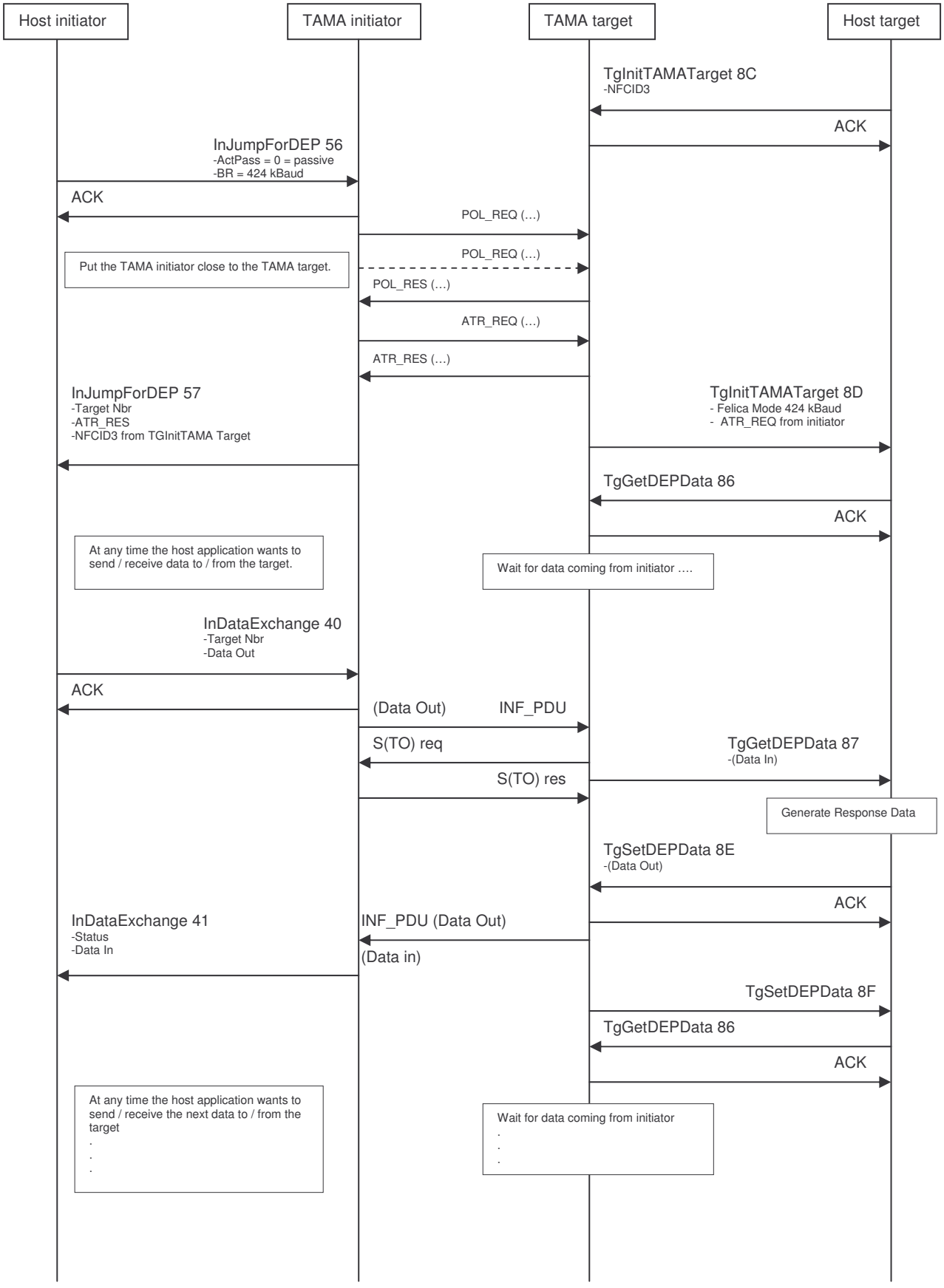
NFC active-passive peer-to-peer communication at 106 kBaud



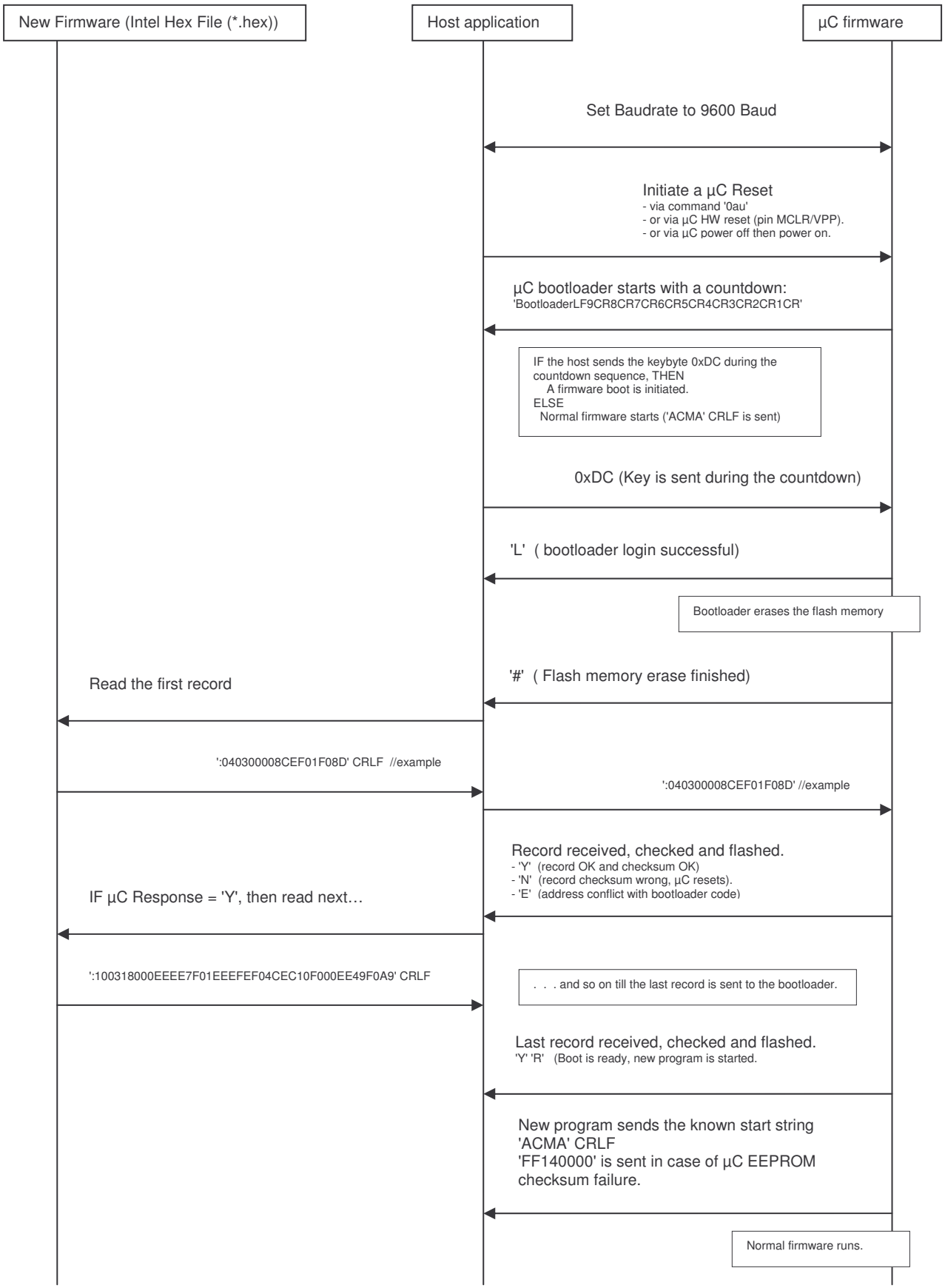
NFC active-active peer-to-peer communication at 212 kBaud



NFC active-passive peer-to-peer communication at 424 kBaud



6.5 µC bootloader scenario:



7. Stand-alone access control with Wiegand™ interface

7.1 Wiegand™ reader highlights

- The Wiegand™ reader can act with or without a Wiegand™ host, that means the reader can be set into a real stand-alone mode without any host.

- In stand-alone mode, Mifare® tags are checked and if access is allowed the reader sets autonomously a digital output pin for 1,5s to high. This can be used for electromagnetic-door opener for example.

- In *stand-alone mode* the Mifare® key and a fixed block identifier are checked to get access.

- In Wiegand™ *host mode* the 4 byte serial number or 4 byte predefined block data are transmitted via the Wiegand™ interface to the host.
  If block mode is chosen, own individual user numbers for each card can be defined.
  In host mode the host checks the received data and sets the reader "Wiegand™ enable line" if access is allowed.

- To increase the security, in any case the Mifare® key and the fixed user block identifier are checked before any data are sent to the Wiegand™ host. This check is independent if stand alone mode / Wiegand™ host mode, or if card serial number / block mode is chosen.

- The Wiegand™ reader can be personalised by normal Mifare® tags with so called Mifare® master cards (no PC or notebook equipment needed for reader personalisation).

- With master cards the following reader settings can be modified (non-volatile of course):

  - Change the master key for login into master cards.
  - Change the user key for login into user cards.
  - Select UID Mode: Card serial number or block data are sent via the Wiegand™ interface.
  - Select the sector and block number for the Wiegand™ storage space inside user cards.
  - Change the host / stand alone mode: With or without Wiegand™ Host.

- The data to the Wiegand™ host are transmitted according to the Wiegand™ standard, specified by the Securities Industry Association (SIA).
  The 34-bit Wiegand™ protocol is used.
  (four byte serial number or four byte block data + 2 parity bits = 4 x 8 + 2 = 34 bits)

- Mifare® standard tags are used for master cards (personalisation) and for user cards.

- The reader is able to communicate with the user via LED blink sequences to report the status of the access control or of the personalisation procedure.

7.2 Wiegand™ reader workflow

The standard µC firmware upon version 2.0 and higher includes the access control functionality. The user can switch from the normal reader behaviour to the stand-alone access control functionality by changing one byte in the µC EEPROM. After changing of this "Standalone" Flag to 1, all relevant access control EEPROM contents are set to access control default value.

Once the reader is set into access control mode it begins to scan for mifare standard tags. First the reader tries to log into user tags, if this fails the master card format will be tested. If the detected tag does not match neither to the user card nor to the master card format an error blink code will be generated on the red LED to inform the user. Then the reader scans for new mifare tags, tries to log into user cards, if fails then log into master cards and so on…

If the scan of the user tag and login was successful then the reader sets autonomously a digital output pin and the green LED for 1,5s to high. This can be used for electromagnetic door opener for example.
- In stand alone mode this happens immediately.
- In Wiegand™ host mode this happens after the host drives the "Wiegand™ enable line" to low for a while.

To avoid an endless door opener activation, a tag is read only once.

7.3 EEPROM organization in case of access control mode

EEPROM Register	*Description*
0x0000	rfu.
0x0001 … 0x0004	EEPROM login pincode.
0x0005	Unique Reader ID for party line commands. **Default ID = 0x01**.
0x0006	**Stand-alone On (0x01)** / Off (0x00). Default is stand-alone off (0x00).
0x0007 … 0x0009	rfu.
0x000A	Mifare® key type for Mifare® authentication (key number 0x00).
0x000B … 0x0010	Mifare® key (6 Byte) for Mifare® authentication (key number 0x00).
0x0011	Mifare® key type for Mifare® authentication (key number 0x01).
0x0012 … 0x0017	Mifare® key (6 Byte) for Mifare® authentication (key number 0x01).
…	…
0x00D5	Access control adjustments RFU. (key number **0x1D**).
0x00D6 … 0x00DB	Mifare® **Access control adjustments** (key number **0x1D**).
0x00DC	Mifare® key type for Mifare® **user card** authentication (key number **0x1E**).
0x00DD … 0x00E2	Mifare® key (6 Byte) for Mifare® **user card** authentication (key number **0x1E**).
0x00E3	Mifare® key type for Mifare® **master card** authentication (key number **0x1F**).
0x00E4 … 0x00E9	Mifare® key (6 Byte) for Mifare® **master card** authentication (key number **0x1F**).
0x00EA … 0x03FD	User free read/write data area.
0x03FE	EEPROM data valid flag.
0x03FF	EEPROM checksum Byte.

7.3.1 EEPROM organization: Access control adjustments

The key number **0x1D** is re-used for the **access control adjustments**.

D5	D6	D7	D8	D9	DA	DB
RFU	**Version** 1.0	**Mode:** 0 = UID 1 = Block	**User Block** Nbr. (0x01…0x3F)	**Host** 0 = no host 1 = with host	RFU	RFU

Version:
Access control workflow version number. In future this byte can be used for distinction between different software versions concerning access control.

Mode:
Card serial number or block data are sent via the Wiegand™ interface to the host
mode = 0x00 = UID is sent.
mode = 0x01 = Block data are sent.

User Block:
Block number for the Wiegand™ storage space inside user tags.
PN531 specific calculation formular of the block nbr. as function of sector:

first block nbr of a sector(beginning from 0) = sector(beginning from 0) x 4

Host:
Host mode = 0x00: Real stand-alone access control mode without any host.
Host mode = 0x01: The Wiegand™ host decides if access is allowed or not. For this the host sets the reader Wiegand™ enable line to low after the received data are validated. The host has to meet some timing requirements for switching of the enable line:

Enable line = High = no access = default because of the reader pull up resistor.
Enable line = Low = access is now allowed (the reader sets the output pin for electromagnetic door opener for example to high)
After the last Wiegand™ bit is sent from the reader to the Wiegand™ host, the reader checks after a **delay of 100ms** if the enable line is driven to low. So the host has **to ensure** that at this time the line is correct driven (see in the timing figure below).
For the pulse length there are no special requirements. The low pulse can start after a delay of 20 ms and can take 150 ms long for example.

7.3.2 EEPROM organization: User card login data

The key number **0x1E** is re-used for access control for the **user card** login.

DC	DD	DE	DF	E0	E1	E2
Key Type B	Access control user card key (6 Bytes)					

User Key Type and user login key for user tags.

7.3.3 EEPROM organization: Master card login data

The key number **0x1F** is re-used for access control for the **master card** login.

E3	E4	E5	E6	E7	E8	E9
Key Type B	Access control master card key (6 Bytes)					

Master Key Type and master login key for master tags.

7.4 Master / User Card organization

7.4.1 Master card format (master card personalisation)

Master cards use always the **sector 1** (Blocknumber 4…7).

Block	**00**	**01**	**02**	**03**	**04**	**05**	**06**	**07**	**08**	**09**	**0A**	**0B**	**0C**	**0D**	**0E**	**0F**
4	fixed master ID			RFU	Vers.	Cmd Code	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
5	user key B (new)						RFU	RFU	**Block**	**Mode**	**Host**	RFU	RFU	RFU	RFU	RFU
6	master key B (new)						RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU	RFU
7	Key A						Acc. Cond.: 08 77 8F 69 h				master key B (old)					

Fixed Master ID:
Beside the master key, three fixed bytes (0xA1 0xA2 0xA3) are additional used to identify explicit master cards.

Version:
This byte is reserved for the version number of master cards.

Command Code:
Command = 0x00: For testing the version string is sent via the Wiegand™ interface to the host. e.g.: (Parity1 0x00 0x00 0x00 0x10 parity2).
Then the red LED is blinking twice, and the green LED is blinking twice for test.

Command = 0x01: Update the µC EEPROM with the user and administrator settings.
Command = 0x02: Reset µC EEPROM user and administrator settings to default value.

User key B (new): new user login data.

Block:
User Block number for the Wiegand™ storage space inside user tags. In case of Mifare® standard tags the following block range is allowed: 0x01 ... 0x3F (Block 0x00 is read only).

Mode:
Card serial number or card block data are sent via the Wiegand™ interface to the host.
mode = 0x00 = 4 Byte UID are sent.
mode = 0x01 = 4 Byte Block data are sent.

Host:
Host mode = 0x00: Without Wiegand™ host. Real stand-alone access control mode.
Host mode = 0x01: With Wiegand™ host.

Master key B (new): new master login data.

Acc. Cond.:
During personalisation of the cards, fixed access conditions are used for the whole access control handling (08 77 8F 69 h). Key B is always used.

7.4.2 User card format (user card personalisation)

User card format (sector / block is **selectable**).
The following block number is only an example. Block 0x26 (within Sector 9):

Block	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0x24																
0x25																
0x26	fixed user ID			User defined Block ID												
0x27	Key A						Acc. Cond.: 08 77 8F 69 h				user key B					

Fixed User ID:
Beside the user key, three fixed bytes (0xB1 0xB2 0xB3) are additionally used to identify explicit user cards.

User defined block ID:
In this case the user can define 4 individual byte. In Wiegand™ host mode these 4 byte are transmitted via the Wiegand™ interface to the host for further checking.

Acc. Cond.:
During personalisation of the cards, fixed access conditions are used for the whole access control handling (08 77 8F 69 h). Key B is always used.

Remark:
Because of the fact that the block is selectable, it is possible to personalise a tag so that several access control blocks are used for several Wiegand™applications (up to 63). With this trick it is possible to create one tag which works with different access control readers and access levels.

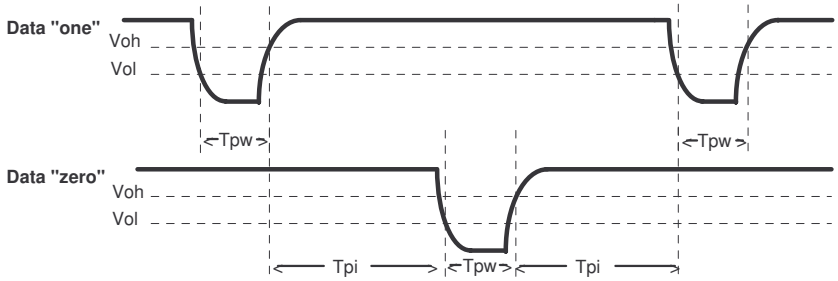
7.5 Reader EEPROM default settings

Default settings are stored in the reader at delivery. The reset operation initializes the reader to the default settings as well.

Setting	Value	Description
Master key	FF FF FF FF FF FF h	Default key (always keytype B is used)
User key	FF FF FF FF FF FF h	Default key (always keytype B is used)
Mode (UID or block data)	00 h	UID is sent to the host
User block	04 h	User block number
Host	00 h	No Wiegand™ host, real stand alone mode

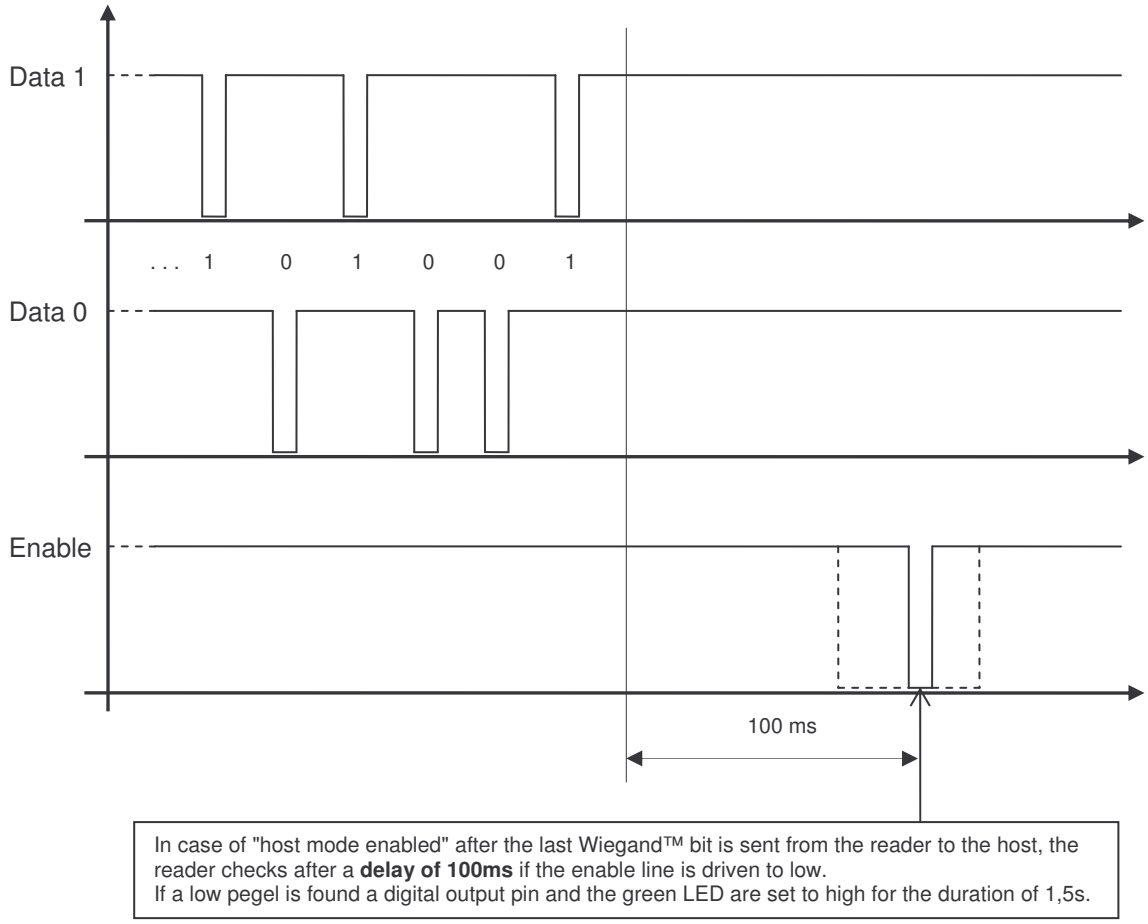
7.6 The Wiegand™ interface

7.6.1 Timing of Wiegand™ data lines



Symbol	Value	Description
Tpw	100 µS	Pulse Width Time
Tpi	1ms	Pulse Interval Time
Voh	4.0V…5.5V	Voltage out high
Vol	0.0V…1.0V	Voltage out low

7.6.2 Timing of Wiegand™ enable line (with host mode enabled)



In case of "host mode enabled" after the last Wiegand™ bit is sent from the reader to the host, the reader checks after a **delay of 100ms** if the enable line is driven to low.
If a low pegel is found a digital output pin and the green LED are set to high for the duration of 1,5s.

7.6.3 Wiegand™ bit stream example

The 34-bit Wiegand™ protocol is used.
(four byte serial number or four byte block data + 2 parity bits = 4 x 8 + 2 = 34 bits)

Following is an example given for the Byte sequence 04 60 22 A8 h:

```
     0   4       6   0       2   2       A   8
1  0000 0100   0110 0000  0010 0010  1010 1000  0
P1 EEEE EEEE   EEEE EEEE  OOOO OOOO  OOOO OOOO  P2
```

Hexadecimal
Wiegand™ bit stream
Description

P1:	First, or even parity bit.
P2:	Second, or odd parity bit.
E:	Bits for calculation of even parity.
O:	Bits for calculation of odd parity.

7.7 Wiegand™ reader LED blink code

The reader is able to communicate with the user via LED blink sequences to report the status of the access control workflow or of the personalisation procedure.
Meaning of the LED blink codes:

Green LED blink code:

Blink Code (LED on)	Description
2	Action successful
1.5s permanently on.	Pin (P7) for door opener is set to high als well.

Red LED blink code:

Blink Code (LED off)	Description
Permanently on	Reader is in idle state.
1	RFU
2	Found card does not match with the fixed user or master card ID.
3	Wiegand™ host does not accept this correct card at the moment. (condition: host flag = 01).
4	Found card does not match with the stored key or key type.
5	Internal error: Internal µC EEPROM checksum failed or internal ringbuffer is not completely filled.

8. References

ISO/IEC 14443 Part 1- 4, *Identification cards – Contactless integrated circuit(s) cards - Proximity card(s)*

ISO/IEC 18092, Standard ECMA-340, *Near Field Communication – Interface and Protocol (NFCIP-1)*

PN531 User Manual, *UM0501-xx* (or newer)

Mifare® standard Card IC MF1 IC S50

Mifare® standard 4kByte Card IC MF1 IC S70

Mifare® ultralight MF0 IC U1